

SwithS(Study with Skku): Study integration platform for SKKU Students

IlGeon Kim¹, YiJi Kim², RoWon Chung³, and KyungJune Shin³

¹ Department of Fineart, Sungkyunkwan University

² Department of Systems Management Engineering, Sungkyunkwan University

³ Department of Mathematics, Sungkyunkwan University

Abstract. Our school, Sungkyunkwan University provides a variety of study and mentoring(tutoring) programs to support our academic and personal growth. Also, students actively engage in these offerings. However, despite the breadth of these programs, the school lacks a unified platform for study program management. The most frequently used platform has been Naver Cafe[3][4], but it primarily serves as a tool for monitoring program outcomes and does not provide essential features during learning process. Additionally, when students are unable to participate in school-run programs, there is currently no convenient study platform available for individual use during the study group recruitment process. Therefore, this paper proposes a learning platform exclusively for students at Sungkyunkwan University, called SwithS(Study with SKKU). This platform enables students to either recruit or join study groups while also providing essential features such as study join and resource sharing to support effective learning. We believe that this platform will not only enhance the academic capabilities of our students but also improve overall efficiency within the school-run programs.

keyword: study community timetable interview

1 Introduction

“SwithS” is a web-based study platform for students at Sungkyunkwan University. According to our survey, about 73% of 82 respondents, or 60 people, said they had participated in the school’s study/mentoring/teaching program. In addition, 49 people answered “Yes” to the question, “If you have a new platform to manage study/mentoring, do you think you can use it more conveniently than the existing methods (Naver Cafe, documentation and submission)?”. 38 people said they had difficulty recruiting and applying, with 38 responding that they had sought a study at Everytime. In addition, 66 people expressed their intention to use a study platform dedicated to students at Sungkyunkwan University. In this respect, the main functions of the platform will be of sufficient help to students. Details of the platform include:

1. Register, Login
Use skku email authentication.

2. Register for a study
3. Search
Search by search word, category.
4. My study room
Provide announcements and study materials functions for each study.

We will provide more detailed explanations of these functions in the following sections.

Our project leverages various technologies to power its functionality. For the frontend, we've opted for React as our framework of choice, enabling dynamic and responsive user interfaces. On the backend, we've employed Spring, providing a robust and scalable foundation for our server-side operations.

In terms of databases, we're utilizing MySQL as our relational database, ensuring structured and efficient data storage and retrieval. Additionally, we're leveraging Redis as our NoSQL database, facilitating fast and flexible data handling, particularly for caching and real-time applications.

The project is restricted to Sungkyunkwan University students, with limited access for non-logged-in users. It allows users to find and view study groups based on major, but features like study group management and time management are still incomplete. Multiple users can access simultaneously, and access is restricted to school email addresses. However, mobile access or app support is not implemented due to time limitations.

2 Design for the Proposed System

2.1 Overall Architecture

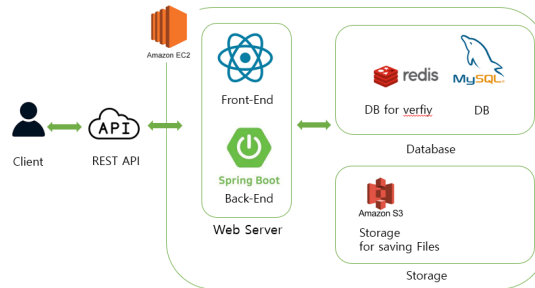


Fig. 1: Overall System Architecture

Our overall system architecture is structured on AWS EC2, enabling a robust cloud infrastructure. Leveraging this, we've organized our databases, frontend, and backend components.

The frontend is constructed using React, providing a dynamic and interactive user interface, while the backend relies on Spring services, ensuring a powerful foundation for server-side operations.

For critical processes such as email authentication, we've utilized Redis to store verification codes for a limited duration of 5 minutes, offering enhanced security and efficiency in the authentication process.

Handling images and files is streamlined through the use of the S3 service. We've created dedicated buckets for our service, granting access and storing images and files within these buckets. This approach significantly reduces the load on our databases, optimizing performance.

This architecture allows for scalability, security, and efficient management of resources, utilizing AWS EC2 as the backbone to orchestrate the various components, enhancing the overall reliability and performance of our system.

2.2 Skills

1. SMTP

We employed SMTP services for our email transmission. By setting up a Google SMTP account, we enabled the system to dispatch emails to our users. Upon sending these emails, authentication codes are stored in Redis for a duration of 5 minutes, facilitating the users' email verification process.

2. JWT

We proceed with user authentication using JWT technology. When a user sends a login request, we verify if the user is a registered member of our service. Upon successful verification, the user is provided with an access token and a refresh token. The access token remains valid for one hour. Users include these tokens in the headers when making API requests, allowing the server to identify the user making the request securely. Furthermore, users need to obtain these tokens to access certain pages, such as creating new study.

3. S3 Storage

To handle the storage of images and files efficiently, we utilize S3 technology. Instead of directly storing these files in the database, which could significantly burden the system, we opt to create dedicated buckets within S3 to store the files separately. Consequently, in the database, we store the file names as strings, allowing users easy access to the files.

This approach significantly reduces the load on our database, ensuring smoother performance while also providing a convenient way for users to access their files stored within the buckets.

4. Selenium

We utilized Selenium technology to perform crawling of Everytime's timetable information. Since the information that we need didn't require login credentials, I employed Firefox as the browser and the GeckoDriver for this task.

2.3 Challenges

During the initial stages of configuring S3 access, I encountered significant challenges. Initially, I generated and utilized access keys from Amazon Web Services (AWS). However, an inadvertent mistake led to the exposure of these access keys in the Git repository, posing a serious security threat.

Further exploration and study of S3 highlighted the critical importance of security concerns. Promptly, the access keys were deleted, and a more secure approach was sought.

Consequently, an IAM (Identity and Access Management) account was created to grant full permissions specifically for S3. This strengthened our security measures. To prevent sensitive information from lingering in Git, the `git rm` cache command was executed to remove all exposed data.

To facilitate smooth front-end development, a DEV API server was established. While typical APIs functioned well, issues arose within the crawling algorithm. The problem stemmed from differences between the DEV server environment (Linux) and the working environment (Windows), resulting in varied installation and execution processes for the crawler. Despite encountering several errors, including the need for additional options in the CLI environment, eventual modifications led to success.

When utilizing an EC2 as the API server, attempts to use the free-tier EC2 micro instance with 1vCPU and 1GB RAM proved insufficient, causing server crashes. Consequently, opting for a paid medium-specification version became necessary, where money support was applicable until November but ceased from December onward. Consequently, deployment processes were significantly streamlined compared to the initial plan due to these changes.

Upon merging into the Develop branch via PRs, automatic deployment via Github Actions was configured. Employing a Self-hosted runner with EC2 setup, automation for triggering deployment through a bash file was established. However, an issue arose where Github Actions recognized executed Java processes, initiated at the end of its run, as orphan processes, terminating them. This was resolved by modifying the bash file, adjusting permissions, and executing with `'sudo'`.

3 Implementation

3.1 User

1. Frontend

On the registration page, users can sign up by providing their name, Sungkyunkwan University email, and password. Users must verify ownership of their email by entering a verification code, which is sent to the provided email address. Upon successfully completing these verifications, users can finalize the registration process.

On the login page, users can log in by entering their Sungkyunkwan University email and password. Additionally, there is a button that allows users to navigate to the registration page. Upon successful login, the server issues a token.

2. Backend

For user registration, when a user clicks the email verification button, a POST API call is made, utilizing the SMTP service to send an authentication code to the provided email address. This authentication code is stored in Redis for 5 minutes. Subsequently, the API responsible for checking the correctness of the authentication code verifies both the user's email information and the code; success occurs only when both are accurate.

Regarding the login process, when a user sends a login request, the email and password are transmitted to the server. If the user's identity is confirmed, they receive access and refresh tokens. These tokens allow the user to verify their information and access specific pages.

3.2 Main page

1. Frontend

At the top of the main page, the header includes a logo, a search bar for entering search queries, and a login button. When the user clicks the login button, they are directed to the login page, and upon successful login, the login button transforms into a person-shaped icon. Additionally, a new study creation button that allows users to navigate to the post page is displayed. Clicking person-shaped icon reveals a dropdown containing buttons for accessing the study room, uploading timetables, and logging out.

Below the header, there is a sequence of sections, including a banner image section, the latest studies, as well as sections allowing users to search for studies by category. Latest studies section displays the five most recent posts, which users can click on to navigate to the detailed study pages. Category-specific studies can be explored by clicking on category buttons, leading users to dedicated pages for each category.

2. Backend

We have an API that fetches the currently available categories and retrieves the most recently created study among the created studies. Additionally, the API provides the ability to explore studies based on these categories. This functionality allows users to stay updated on the latest studies and easily navigate through the diverse categories offered on our main page.

3.3 Post Page

1. Frontend

The post page is a page for creating new study recruitment posts. Users can input the study's name and description as text, and select the recruitment type, number of recruits, and major using dropdown menus. The recruitment period and study duration are input using a calendar library. When the user presses the complete button, the posting is finalized.

2. Backend

On the post page, we utilize an API that posts clubEntity. This API takes inputs containing information such as titles, content, category, number of recruits, major, start of recruitment, end of recruitment, start of progress, and end of progress. The provided information is then posted to ClubEntity, allowing users to share details about their club, including recruitment timelines and other essential details. This functionality streamlines the process of creating and posting information about various clubs on our platform.

3.4 Search, Category Page

1. Frontend

When a user enters a search query in the header's search bar, they are redirected to the search page. Similarly, clicking on a category button in the category section takes the user to the category page. These pages display recruitment posts filtered by the respective search term or category, presented in card format. The cards can be filtered using a category select box. For study cards, information such as the recruitment status, study name, study duration, and study description is included. Clicking on these cards redirects the user to the detailed study page.

2. Backend

Search functionality on our platform includes three main options: searching by category, searching by keyword, and utilizing the API that performs a comprehensive search. Users can look up ClubEntity to retrieve all clubs containing specific keywords. Additionally, there's an option to narrow down searches by category, providing a more focused and tailored exploration experience. This versatile search system ensures users can easily find clubs of interest based on their preferences and criteria.

3.5 Study detail Page

1. Frontend

On the study detail page, the user can view the information entered when creating the study recruitment post. Clicking the join button adds the study to the My study room page.

2. Backend

On this page, upon pressing the Join button, the study information and user information are automatically added to the user club Entity. This feature facilitates the management of users who have joined the club, providing a centralized view of club memberships.

Furthermore, there is an API designed to compare the timetables of individuals currently in the club with the timetable of the user who signs up. This comparison helps determine suitable time slots for conducting the study, enhancing the overall scheduling and coordination of club activities. This seamless integration of user and study information streamlines the process of joining clubs and ensures efficient planning of study sessions based on participants' availability.

3.6 My study room Page

1. Frontend

The my study room page displays all the studies that the user has added in card format. Clicking on each card navigates to a page where users can view announcements and materials specific to each study.

2. Backend

Utilizes an API that leverages UserClubEntity to import the details of the study to which the user subscribes. This process involves referencing the UserClubEntity, enabling the seamless integration of user subscriptions with relevant study information.

3.7 Notice Page

1. Frontend

The notice page displays announcements for each study in a list format. When the user clicks the upload notice button, they can input the announcement title and content. Upon clicking the registration button, the announcement is immediately submitted. Clicking on an announcement in the list will display detailed information about that particular notice.

2. Backend

An API for uploading notices to PostEntity streamlines the process of sharing important study-related information. It accepts inputs such as titles, content, userid, and clubid, facilitating efficient dissemination of notices within the platform.

The API designed for retrieving notices based on the current study's clubid ensures users can easily access relevant announcements and updates. This functionality enhances user experience by providing targeted and study-specific information.

Additionally, the voting API for setting meeting times enhances collaborative decision-making within study groups. By allowing members to collectively vote on meeting times for the next session, this API promotes a democratic approach to scheduling, fostering cooperation and coordination among study group members.

3.8 Resource Page

1. Frontend

The materials page displays all the materials for each study. When the user clicks the upload resource button, they are directed to a place for uploading materials. The material is registered with the same name as the file, and clicking on a material in the list allows the user to immediately download the material to their local device.

2. Backend

When a user clicks on a file name, the server internally combines it with

the bucket address and sends a request to the S3 bucket for file download. Through this request, users can download the file to their local device. Additionally, by sending a GET request to the S3 bucket, users can view the list of files within the bucket. For file uploads, upon receiving files of type 'multipart/form-data', a POST request is sent to the S3 bucket address, allowing the user-uploaded files to be stored in the actual bucket. Files exceeding a certain size limit will not be uploaded.

3.9 Timetable upload Page

1. Frontend

The timetable upload page is a page where users input their EveryTime URL for timetable upload. If the URL is not in the correct format, the registration fails.

2. Backend

The Timetable upload API, powered by Selenium, plays a crucial role in automating the extraction of timetable information based on the provided Everytime URL. This API efficiently navigates through the Everytime website, retrieves the relevant data, and compiles essential timetable details for the user.

By leveraging Selenium, the API automates the interaction with the Everytime platform, mimicking user actions to access and gather timetable information seamlessly. This automation not only enhances user convenience but also ensures accuracy and consistency in retrieving the required data from Everytime URLs.

4 Planning in Detail

4.1 Role

IlGeon Kim will be the team leader and will be in charge of project management. YiJi Kim will be dedicated to ui/ux, and will do page implementation. RoWon Chung will do deployment and api development. KyungJune Shin will do db design and api development.

4.2 Actual Schedule

The development has been completed up to Week 15. Anticipating significant expenses during the deployment process, we opted to conduct the deployment in Week 16, coinciding with the Final Project Presentation week.

5 Evaluation, Limitations and Discussions

Usability Test and Answers Internal evaluation tests were conducted with a focus on the following aspects, and the answers to each question are as follows.

Task	Assignee	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
Individual study and project setting	Every Members												
Figma design	Frontend												
DB design & back API creation	Backend												
Web Page Development	Frontend												
API Development	Backend												
Testing and optimization	Every Members												
Deployment	Infra												

Fig. 2: Actual Schedule

1. Project Purpose Related

- (a) Is it exclusive to Sungkyunkwan University students?
 - i. Are non-logged-in users restricted from accessing information?
 - Yes, information is restricted for non-logged-in users. In addition, we limited the scope of disclosure by adding a user authentication process during the API call process.
- (b) Is it suitable for recruiting study groups or project members?
 - i. Can users find study groups or projects based on major or topic?
 - Yes, users can search for any study they want based on their preferred major, field, or study name.
 - ii. Can users view recent created studies?
 - Yes, users can view recently created studies. By providing recently created studies, users can understand recent trends.
- (c) Is it suitable for managing study groups?
 - i. Can study materials be stored and downloaded?
 - Yes, study materials can be stored and downloaded. We use S3 storage to store and share files such as docs, hwp, PDF, etc. S3 storage can reliably store large amounts of data, and helps to prevent data loss and maintain high availability.
 - ii. Is schedule management possible(Study alarm, Member management, etc)?
 - No, It hasn't been implemented yet. The creation of a page that can confirm the number of people participating in the study group is still incomplete.
 - iii. Is there a time management function for study meetings provided?
 - No, It hasn't been implemented yet. When signing up for a study, it was intended to simplify the function of providing information on the

meeting time in progress or the process of setting the meeting time of the study. The Backend and DB logic have been implemented, but in the last weekly meeting, we received feedback suggesting that it might be beneficial to refer to the UI and logic of When2Meet[5]. I am currently in the process of working on this feedback.

2. Project Functionality Related

- (a) Can multiple users access simultaneously?
 - Yes, multiple users can access and use our platform.
- (b) Is access restricted without a school email?
 - Yes, the authentication process proceeds based on the email address user entered. Only emails from "@skku.edu" or "@g.skku.edu" domains are allowed, and the process of verifying the verification code is also carried out.
- (c) Is it accessible in a mobile environment or through an app?
 - No, it would be nice to support multiple devices. However because of the time limitations, we couldn't implement UI for multiple devices.

Discussions It was a short project for about two and a half months, so we focused on essential functions. Consequently, it's not at the stage where we can deploy the service. Operating the service effectively demands various additional functions, including administrator management, account deletion, and information update. However, our focus leaned towards crucial functions related to the initially planned study platform, such as user scope limitation, study member recruitment, promotion, and study material management. Implementing essential features took more time than expected to deploy Redis and S3 storage, requiring additional effort and time to understand technology and integrate. Consequently, the implementation of additional functions is not yet complete. Moving forward, we also intend to develop additional functions to enhance the overall completeness of the service.

6 Related Works

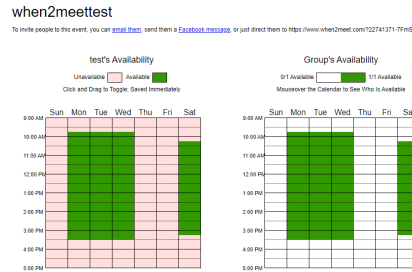
Existing Platforms Participation in learning or tutoring sessions at SKKU involves two main methods: School-Run Programs and Personal Recruiting. School-Run Programs, advertised through official channels and social networks, have fixed recruitment periods, limiting late applicants. Personal Recruiting, often facilitated by platforms like Everytime, allows students to form their own study groups but poses safety concerns due to anonymity. Managing these programs typically occurs through Naver Cafe[3][4], where each study group has a bulletin board for reports, lacking features like schedule management and resource sharing. Alternatively, if no dedicated platform exists, participants resort to email for reporting, relying on external tools for scheduling and resource sharing. HOLA[1] is a developer-oriented platform for finding projects or study

groups, emphasizing recruitment but lacking study process management features. Kingwaowae[2] is a tutoring platform, similar to HOLA, focusing on matching without additional tutoring process support.



Fig. 3: Various Online Platforms

Referenced Platforms When2Meet[5] is an online tool used to find the optimal schedule for meetings or events. It allows users to create a schedule, set up an event, and share a link for participants to choose their availability. Attendees can then vote on preferred time slots, and the organizer can confirm the most suitable time based on the collective responses. During the last weekly meeting, we received feedback suggesting that it might be beneficial to refer to When2Meet while creating the schedule. The UI of When2Meet is quite intuitive, so we decided to adopt it.



(a) When2Meet[5]

7 Conclusions

Sungkyunkwan University students currently seek studies through Everytime or school notices, but organizing studies on messaging apps like KakaoTalk or Naver Cafe limits functionalities. A survey reveals students' significant discomfort with this approach and a recognized need for a dedicated study platform. Our goal is to provide a study platform exclusively for Sungkyunkwan University students, offering features for study recruitment and management. Users proceed with registration using their Sungkyunkwan University email. After registration, they can upload their Everytime timetable on the timetable upload page. Within the main page, users can browse studies by category and participate in desired ones. Through the "My Study Rooms" page, users can view their enrolled studies, access announcements, upload or download study materials within each study room. In the project, we utilized various technologies that we hadn't previously explored. For the user registration process, we employed SMTP services and Redis. Using SMTP through the SwithS account, we conducted verification. The verification code was stored in Redis for a specific duration, and the registration functionality was implemented by comparing user input with the stored code. This allowed us to restrict access to Sungkyunkwan University students. We implemented user authentication using JWT, providing access and refresh tokens to maintain login status. S3 technology was employed by creating dedicated buckets within S3, reducing database load and ensuring convenient user access to stored files, thus efficiently handling image and file storage. Additionally, Selenium was used to crawl timetable information from Everytime, utilizing the Firefox browser and GeckoDriver. If the platform gains popularity within the campus community, it holds the promise of serving a larger user base. The incorporation of Admin features and comprehensive study management functions could significantly contribute to the platform's transformation into a more robust and valuable service for current students on campus. Importantly, there is significant potential for its development as an official study platform for Sungkyunkwan University.

References

1. Hola Homepage, <https://holaworld.io/>. Last accessed 9 Dec. 2023.
2. 김과외 Webpage, <https://kimstudy.com/main>. Last accessed 9 Dec. 2023.
3. "성균튜터링 카페." 네이버 카페, <https://cafe.naver.com/skkututoring>. Last accessed 9 Dec. 2023.
4. "성균관대학교 소프트웨어학과 카페." 네이버 카페, <https://cafe.naver.com/onlyforskkusoftware>. Last accessed 9 Dec. 2023.
5. When2Meet, <https://www.when2meet.com/>. Last accessed 9 Dec. 2023.